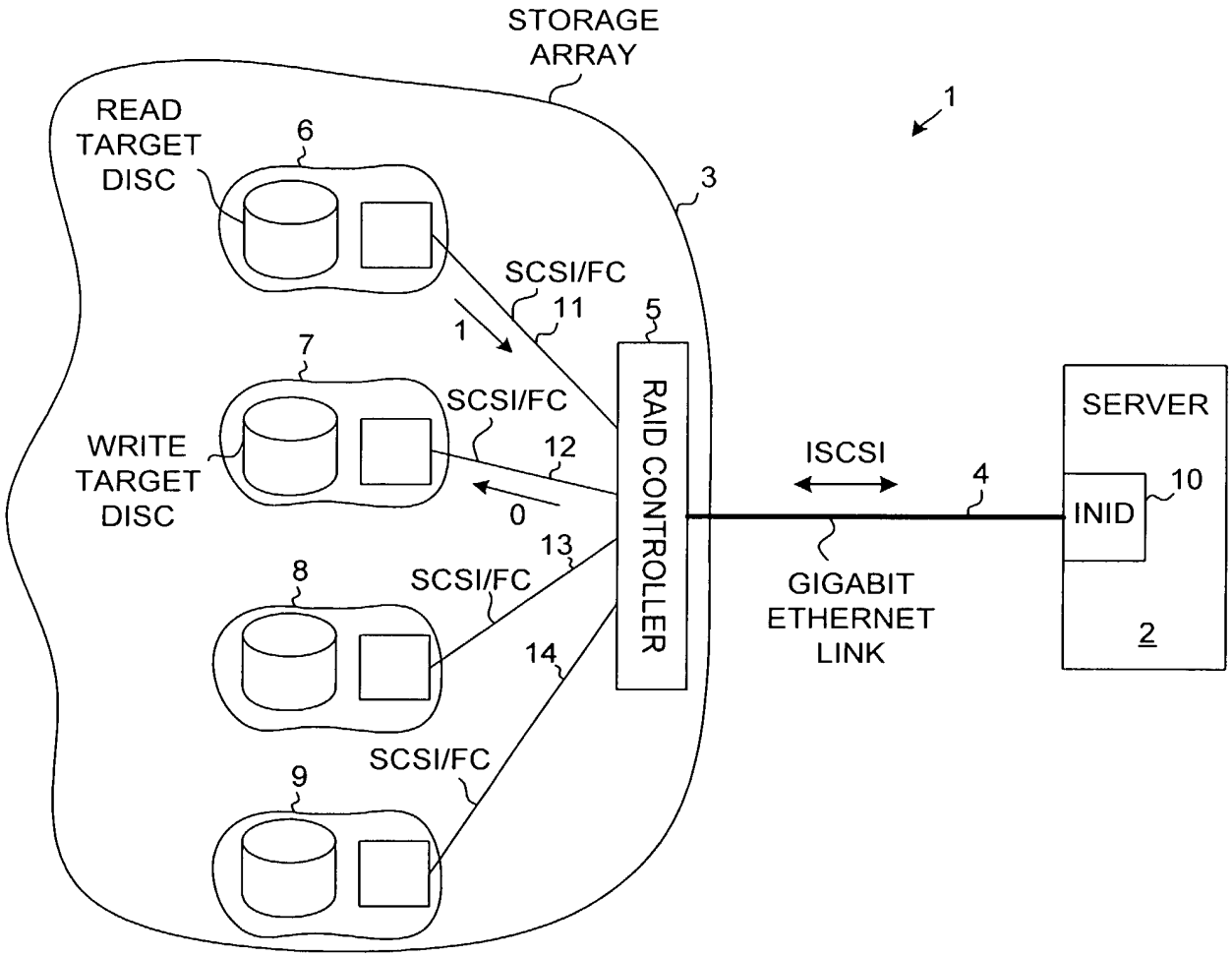

STORAGE ARRAY
READ TARGET DISC
6
3
1
SCSI/FC
11
5
7
SCSI/FC
WRITE TARGET DISC
12
0
RAID CONTROLLER
ISCSI
4
SERVER
10
INID
13
8
SCSI/FC
GIGABIT ETHERNET LINK
14
2
SCSI/FC
9

FIG. 1

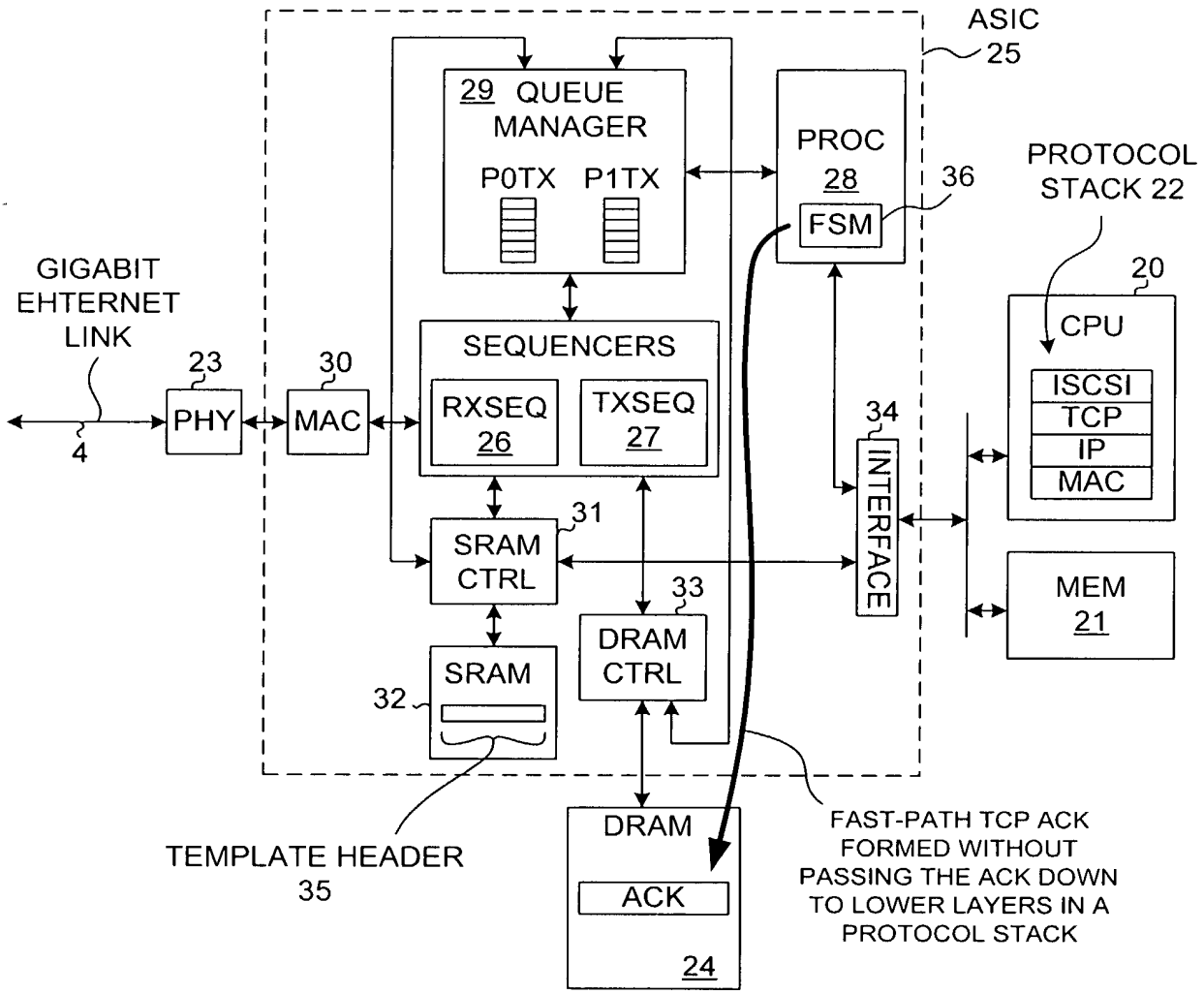
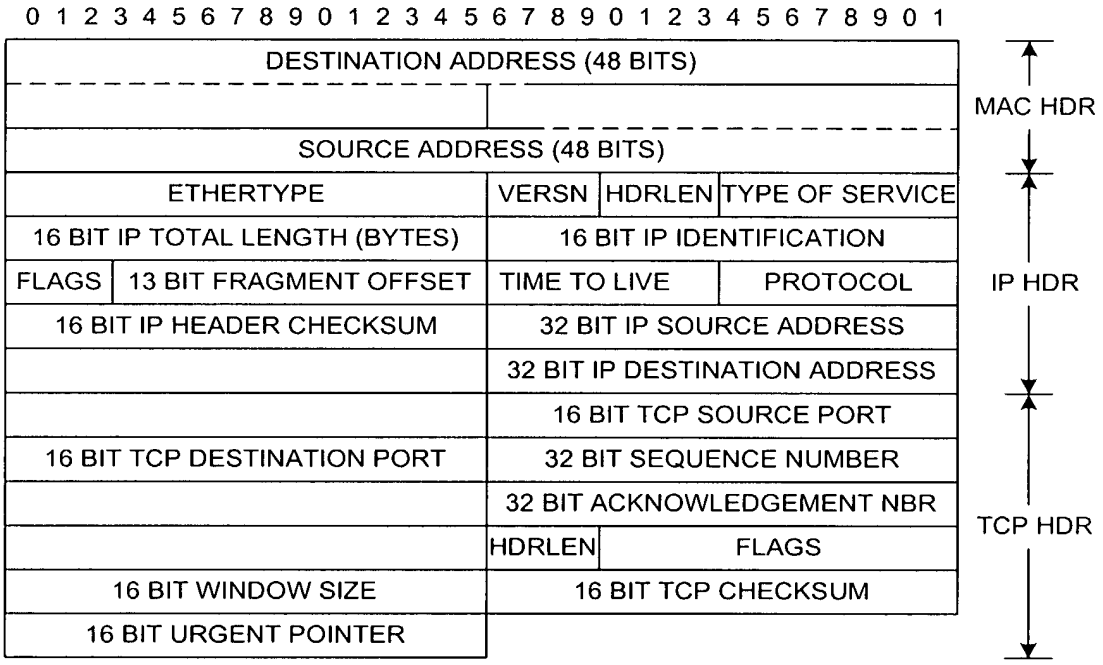

ASIC
25
29 QUEUE MANAGER
P0TX P1TX
PROC
28
FSM
36
PROTOCOL STACK 22
20
CPU
ISCSI
TCP
IP
MAC
GIGABIT EHTERNET LINK
4
23
PHY
30
MAC
SEQUENCERS
RXSEQ 26
TXSEQ 27
34
INTERFACE
31
SRAM CTRL
33
DRAM CTRL
MEM 21
SRAM
32
TEMPLATE HEADER 35
DRAM
ACK
24
FAST-PATH TCP ACK FORMED WITHOUT PASSING THE ACK DOWN TO LOWER LAYERS IN A PROTOCOL STACK

FIG. 2



TEMPLATE HEADER FORMAT

FIG. 3

```
/*---------------------- BEGINNING OF CODE THAT DOES ACK TX PROCESSING ----------*/

AX_DAP:                 /***** AX_DAP *****/
        /*
         * DO ACK PROCESSING.
         *
         * THE RCV DELACK TIMER HAS EXPIRED.
         * FORMAT AND SEND AN ACK.
        */

        /* TI_LENL6            EQU   L6                                        */
        /* LNBPL8              EQU   L8      LARGE INIC BUFFER PTR             */
        /* TCP_CSUML11         EQU   L11                                       */
        /* SYS_SCR             EQU   SYS_SCR                                   */

        /* GET A DRAM BUFFER TO PUT PCI PAYLOAD IN                             */
        JSR     GETLDBUF;                           /* (LNBPL8)              */
        TEST    LNBPL8, JCF ZERO DAP0_1;            /* GO ON IF GOT A BUFFER */
         /* NO BUFFER - SEND EVENT AND TRY LATER    */
        MOVE    GR0 EX_SACKC,
        JSR     SETOPEVNT;                          /* (THWD0L12)            */
        JMP     XFSM_EXIT;

DAP0_1:
        /* SETUP CANNED HEADER. */
        CLR     TI_LENL6,
        JSR     SETCANNEDHDDR;   /* (TI_LENL6, TOTLENL10, TCP_CSUML11) */

        /* DMA TEMPLATE HDDR FROM SRAM TO DRAM BUFFER */
        /* SET   CHKSUM INTO HEADER */

        ADDL    ADDR_REGB TCBSRAML5 STCB_TEMPLATE+TPL_TCPCSUM|ADDR15;

        /* READ TEMPLATE HEADER FROM SRAM */
        ANDL    SYS_SCR TCP_CSUML11 H'FFFF';
        SHFTR   TCP_CSUML11 C16, LIT_TO_ADDR_REGA STCB_TCPCB+TCB_SHFLAGS;
        ADD     TCP_CSUML11 SYS_SCR, JCF ALU_B16 '$ + 2';
         INCR   TCP_CSUML11;                        /* ADD IN CARRY     */
         BTEST  SRAM1 TSF_VLANC, JCT ZERO '$ + 2';  /* GO NO VLAN TAG */
```

FIG. 4A

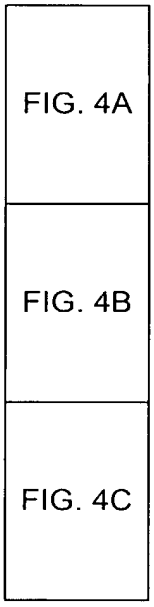
```
ADD      ADDR_REGB VLAN_TAG_SZEC;
XOR      TCP_CSUML11 MINUS_1, WSRAM2_XPOSE;

/*
 * SET    FRAME LEN INTO TEMPLATE HEADER FOR MAC SEQR.
 * FRAMELEN = TEMPLATE HDR LEN - 2
 */

MOVEL   ADDR_REGA STCB_TEMPLATE+TPL_TMPLTLEN;
/* POINT TO TEMPLATE LENGTH */
MOVE     CR0 SRAM2, LIT_TO_ADDR_REGB STCB_XMTBYTCNT;
COMP     CR0 MIN_FRAME_LEN, JCT LT '$ + 3';

/* MAKE SURE FRAME IS MIN LENGTH */
SUBL     NULL CR0 XMT_HDDR_SIZE_SRAM, WSRAM4;       /* TEMPLATE HDR - 2 */
JMP      '$ + 2';
MOVEL    SRAM4 ETHER_MIN_TU;     /* MIN ETHER FRAME LEN = 60 (+CRC) */
ADD      CR0 SIZEOF_XMITHDR+7C; /* PREPARE TO ROUNDUP XFER SIZE */
MOVE     RAM_BASE PDDSCPTR, LIT_TO_ADDR_REGB DMA_CMD_WD;
ANDNL    SRAM4+ CR0 H'3';

/* PDES->IXFR_SZ ROUNDED TO 8-BYTE BNDRY */
MOVE     SRAM4+ LNBPL8;                                /* PDES->DST_ADDR = */
ADDL     SRAM4 TCBSRAML5 STCB_XMIT_BUFFER;  /* PDES->SRC_ADDR = */

/* XXXDMA ORL CH_CMD CTXT_RPROC CCR_S2D;   */
/* SET UP TO DMA THE ACK FROM SRAM TO DRAM  */
MOVE     Q_CTRL Q_S2DC;                                /* SELECT S2D DMA */
MOVE     Q_DATA PDDSCPTR;
JCF      Q_OP_DONE '$ - 1';
MOVE     CTXT_RTNADL14 PC, JMP PROC_SUSPEND;       /* SUSPEND */
MOVE     RAM_BASE TCBSRAML5;                           /* RESUME */
```

FIG. 4B

```
        /* SET EVENT WD FOR XMT Q */
        BSET    LNBPL8 XMT_TCPIPC;
        MOVE    Q_CTRL Q_XMTPRI1C;
        /* WRITE INTO QUEUE CONTROL REGISTER TO INDICATE THE PRIORITY          */
        /* TRANSMIT QUEUE.                                                    */

DAP0_2:
        MOVE    Q_DATA LNBPL8;
        /* WRITE A POINTER (TO THE ACK NOW IN DRAM) INTO QUEUE DATA REGISTER. */
        /* THIS WRITE CAUSES THE POINTER TO BE PUSHED ONTO THE INDICATED      */
        /* PRIORITY TRANSMIT QUEUE.                                           */
        CLR     ISRL2, JCT Q_OP_DONE DAP0_3;        /* JUMP IF Q'D OK */
        BTEST   Q_FULL Q_XMTPRI1C, JCT ZERO DAP0_2;

DAP0_2A:
        /* DROP FRAME */
        ORL     FRADDL0 LNBPL8 2KB_SMSK;
        JSR     FREELDBUF;                          /* (FRADDL0) */
        JMP     XFSM_EXIT;

DAP0_3:
        BCLR    THWD0L12 HE_RDLACKC,
        JMP     XFSM_EXIT;

/*---------------------------- END OF CODE THAT DOES ACK TX PROCESSING ----------*/
```

FIG. 4C

FIG. 4



KEY TO FIG. 4